

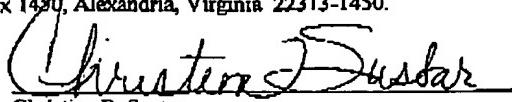
PATENT

JAN 06 2006

MS174305.01 / MSFTP256US

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being faxed to 571-273-8300 on the date shown below to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Date: 1-6-06
Christine R. Sustar**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re patent application of:

Applicant(s): Stephen Peter De Jong *et al.*

Examiner: Brian D. Goddard

Serial No: 09/892,684

Art Unit: 2161

Filing Date: June 27, 2001

Title: PLUGGABLE FORMATTERS

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF

Dear Sir:

Appellants' representative submits this brief in connection with an appeal of the above-identified patent application. A credit card payment form is filed concurrently herewith in connection with all fees due regarding this appeal brief. In the event any additional fees may be due and/or are not covered by the credit card, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1063 [MSFTP256US].

09/892,684

MS174305.01 / MSFTP256US

**RECEIVED
CENTRAL FAX CENTER****I. Real Party in Interest (37 C.F.R. §41.37(c)(1)(i))**

The real party in interest in the present appeal is Microsoft Corporation, the JAN 06 2006 assignee of the present application.

II. Related Appeals and Interferences (37 C.F.R. §41.37(c)(1)(ii))

Appellants, appellants' legal representative, and/or the assignee of the present application are not aware of any appeals or interferences which may be related to, will directly affect, or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims (37 C.F.R. §41.37(c)(1)(iii))

Claims 6, 10, 15, 17-31, 37-48, and 50 have been cancelled. Claims 1-5, 7-9, 11-14, 16, 32-36, and 49 stand rejected by the Examiner. The rejection of claims 1-5, 7-9, 11-14, 16, 32-36, and 49 is being appealed.

IV. Status of Amendments (37 C.F.R. §41.37(c)(1)(iv))

No claim amendments have been entered after the Final Office Action.

V. Summary of Claimed Subject Matter (37 C.F.R. §41.37(c)(1)(v))**A. Independent Claim 1**

Independent claim 1 recites a system that facilitates employment of a pluggable formatter, comprising: a decision module that retrieves a first data structure as a graph of objects for serialization thereof; a plurality of rule sets that define serialization information about data structure types; a serialization selector that determines a rule set of the plurality of rule sets to provide to the decision module based on the data structure type, the decision module populating a second data structure based on the serialization information and providing a pluggable formatter with the second data structure, so that the pluggable formatter can serialize the second data structure to an externalized format defined by the pluggable formatter. (*See e.g.*, pg. 2, line 24 – pg. 3, line 21; pg. 6, line 22 – pg. 14, line 7; *See generally* Figs. 1-5).

09/892,684

MS174305.01 / MSFTP256US

B. Independent Claim 11

Independent claim 11 recites a system that facilitates employment of a pluggable formatter, comprising: a formatter services component that receives a decoded serialized stream from a pluggable formatter and creates a data structure as a graph of objects for deserialization of the decoded serialized stream; and an object manager that tracks data in the decoded serialized stream and determines forward references to additional data to provide fixups to the data structure upon receipt of the additional data. (*See e.g., pg. 2, line 24 – pg. 3, line 21; pg. 6, line 22 – pg. 14, line 7; See generally Figs. 1-5*).

C. Independent Claim 32

Independent claim 32 recites a method for deserialization of a stream into a graph of objects, the method comprising: receiving a decoded serialized stream from a pluggable formatter; retrieving from the decoded serialized stream an object from a graph of objects; determining an object type for deserialization; instantiating an uninitialized instance of the object type; selecting a serialization rule set from one of a user defined rule set in the object, a user defined rule set in another object and a default rule set; and populating the uninitialized instance of the object type based on the selected serialization rule set. (*See e.g., pg. 14, line 21 – pg. 17, line 4; See generally Figs. 6-9*).

D. Independent Claim 49

Independent claim 49 recites a system that facilitates making formatters pluggable, the system comprising: means for providing serialization information of an object from a graph of objects to a pluggable formatter, so that the pluggable formatter can serialize the object in a selectable externalized format; means for tracking deserialization of the object outside the pluggable formatter; means for instantiating an uninitialized instance of an object type, so that the object information can populate the object type; and means for performing fixups on the objects due to forward references. (*See e.g., pg. 2, line 24 – pg. 3, line 21; pg. 6, line 22 – pg. 14, line 7; See generally Figs. 1-5*).

09/892,684

MS174305.01 / MSFTP256US**VI. Grounds of Rejection to be Reviewed (37 C.F.R. §41.37(c)(1)(vi))**

A. Claims 1-5, 7-9, 11-14, 16, 32-36, and 49 stand rejected under 35 U.S.C. §102(e) as being anticipated by Bahrs *et al.* (US 6,292,933).

VII. Argument (37 C.F.R. §41.37(c)(1)(vii))

A. **Rejection of Claims 1-5, 7-9, 11-14, 16, 32-36, and 49 Under 35 U.S.C. §102(e)**

Claims 1-5, 7-9, 11-14, 16, 32-36, and 49 stand rejected under 35 U.S.C. §102(e) as being anticipated by Bahrs *et al.* (US 6,292,933). This rejection should be reversed for at least the following reason. Bahrs *et al.* does not describe each and every element of the subject claims.

For a prior art reference to anticipate, 35 U.S.C. §102 requires that “*each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (*quoting Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)) (*emphasis added*).

The invention as claimed relates to a system and method for facilitating employment of a pluggable formatter in a serialization and deserialization process. The pluggable formatter serializes a graph of objects into a data stream and deserializes the data stream back into an object graph. (*See* pg. 2 line 24 – pg. 3, line 7). In particular, independent claims 1, 11, 32, and 49 recite a similar limitation: a first data structure as a *graph of objects for serialization*. Bahrs *et al.* fails to describe such aspect of the invention as claimed.

The Examiner contends that Bahrs *et al.* describes a graph of objects for serialization with the PlacementListener at col. 16, line 18. (*See* Final Office Action dated August 9, 2005, pages 2, 5, and 6). The Examiner explains that PlacementListener places/contains the overall display of many graphical objects in a certain manner and must maintain such ordered grouping of individual graphical objects for serialization and

09/892,684MS174305.01 / MSFTP256US

deserialization. (See Final Office Action dated August 9, 2005, pg. 6). Appellants' representative respectfully disagrees with such contention.

Bahrs *et al.* discloses the serialization and deserialization of data elements. The serializer receives a data element, replaces its class name string with a code having a smaller size than the class name string to form a modified data element, and serializes the modified data element. In deserialization, the deserializer restores the data element by replacing the indicator with the class name. (See Abstract). Although PlacementListener manages the placement of graphical objects on a display, a *graph of objects* is more than just an arranged display of objects. Bahrs *et al.* may discuss a group of objects, but a *group* of objects is not equivalent to a *graph* of objects. A *graph* of objects is a structure that contains a *graph root*, which is the top object in a graph (See pg. 9, line 28), and subsequent objects that are referenced to other objects in the graph (*i.e.*, forward references and backward references) (See pg. 9, ll. 13-14), as illustrated in Figures 1-5. Bahrs *et al.*'s graphical objects bear no relation to one another beyond the physical positions they hold on the screen. The cited reference fails to describe a *graph of objects* as claimed.

In addition, the graph of objects in appellants' invention is retrieved for *serialization*. Bahrs *et al.*'s PlacementListener handles the placement of objects on a screen (See col. 16, line 26) for user/client viewing. The serialization and deserialization process in the cited reference refers to data elements (See col. 4, line 55), not a *graph* of data elements.

In response, the Examiner contends that Bahrs *et al.*'s "complex data structures" as specified in the title, object oriented programming, and class hierarchies as described in Fig. 6 disclose the graph of objects for serialization. (See Advisory Action dated October 31, 2005, pg. 2). Although a data element to be serialized may include a class name string (See col. 4, ll. 54-56), the cited reference merely describes uses and functions of classes without discussion of a serialization method. The serialization process does not refer to anything more than single data elements.

Bahrs *et al.* also fails to describe an *externalized format* nor a *decoded serialized stream* as a result of serialization, as noted in independent claims 1, 11, 32, and 49. The claimed invention serializes a graph of objects into such types of data streams so that they

09/892,684MS174305.01 / MSFTP256US

are flexible enough to be compatible with external clients. Although Bahrs *et al.* discloses a modified data element, the transformation occurs by replacing the class name string within the data element with an abbreviated code. Then when the data element is deserialized, the code is restored with the original class name. Such replacement of class names with codes is not equivalent to the transformation of an object graph into an externalized format or data stream.

Furthermore, independent claim 34 recites *tracking forward references to other objects within the object* and independent claim 35 recites *repeating, retrieving, determining, instantiating, and populating for each object in an object graph*. The Examiner contends that RequestEvent discloses such aspect. (See Final Office Action dated August 9, 2005, pg. 4). Appellants' representative respectfully disagrees with this contention. RequestEvent indicates that a service is required to process an event. (See col. 16, ll. 59-60). RequestEvent is silent with respect to object graphs, let alone forward references and processing of specific objects within object graphs.

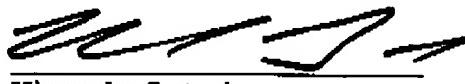
In view of at least the foregoing, it is readily apparent that Bahrs *et al.* does not describe the invention as recited in independent claims 1, 11, 32, and 49 (and associated dependent claims 2-5, 7-9, 12-14, 16, and 33-36). Accordingly, this rejection should be reversed.

09/892,684MS174305.01 / MSFTP256US**B. Conclusion**

For at least the above reasons, the claims currently under consideration are believed to be patentable over the cited references. Accordingly, it is respectfully requested that the rejections of claims 1-5, 7-9, 11-14, 16, 32-36, and 49 be reversed.

If any additional fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP256US].

Respectfully submitted,
AMIN & TUROCY, LLP



Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24th Floor, National City Center
1900 East 9th Street
Cleveland, Ohio 44114
Telephone: (216) 696-8730
Facsimile: (216) 696-8731

09/892,684MS174305.01 / MSFTP256US**VIII. Claims Appendix (37 C.F.R. §41.37(c)(1)(viii))**

1. A system that facilitates employment of a pluggable formatter, comprising:
 - a decision module that retrieves a first data structure as a graph of objects for serialization thereof;
 - a plurality of rule sets that define serialization information about data structure types;
 - a serialization selector that determines a rule set of the plurality of rule sets to provide to the decision module based on the data structure type, the decision module populating a second data structure based on the serialization information and providing a pluggable formatter with the second data structure, so that the pluggable formatter can serialize the second data structure to an externalized format defined by the pluggable formatter.
2. The system of claim 1, the rule set is definable in the data structure.
3. The system of claim 1, the rule set is defined in a third party file.
4. The system of claim 1, the rule set is a default format based on markings in the first data structure.
5. The system of claim 1, the first data structure being an object.
6. (Cancelled).
7. The system of claim 1, further comprising an object ID generator coupled to the decision module, the object ID generator assigns object IDs to each object in the graph of objects.
8. The system of claim 1, the data structure containing information within the data structure that the serialization selector utilizes in determining a rule set.

09/892,684

MS174305.01 / MSFTP256US

9. The system of claim 1, the decision module is integrated into the pluggable formatter.

10. (Cancelled).

11. A system that facilitates employment of a pluggable formatter, comprising:
a formatter services component that receives a decoded serialized stream from a pluggable formatter and creates a data structure as a graph of objects for deserialization of the decoded serialized stream; and

an object manager that tracks data in the decoded serialized stream and determines forward references to additional data to provide fixups to the data structure upon receipt of the additional data.

12. The system of claim 11, further comprising a serialization selector that determines a rule set of a plurality of rule sets to provide to the formatter services component based on the data structure type being deserialized, the formatting services component populates the data structure based on the determined rule set.

13. The system of claim 12, the rule set is definable in the data structure.

14. The system of claim 12, the rule set is in a third party file.

15. (Cancelled).

16. The system of claim 11, the formatter services component reinstatiates each object of a graph of objects and the object manager performs fixups on the objects after the graph of objects have been deserialized.

17-31. (Cancelled).

09/892,684

MS174305.01 / MSFTP256US

32. A method for deserialization of a stream into a graph of objects, the method comprising:

receiving a decoded serialized stream from a pluggable formatter;
retrieving from the decoded serialized stream an object from a graph of objects;
determining an object type for deserialization;
instantiating an uninitialized instance of the object type;
selecting a serialization rule set from one of a user defined rule set in the object, a user defined rule set in another object and a default rule set; and
populating the uninitialized instance of the object type based on the selected serialization rule set.

33. The method of claim 32, further comprising registering the object with an object manager.

34. The method of claim 32, further comprising tracking forward references to other objects within the object.

35. The method of claim 32, further comprising repeating retrieving, determining, instantiating and populating for each object in an object graph.

36. The method of claim 35, further comprising performing fixups on the objects in the object graph caused by forward references.

37-48. (Cancelled).

09/892,684

MS174305.01 / MSFTP256US

49. A system that facilitates making formatters pluggable, the system comprising:
means for providing serialization information of an object from a graph of objects
to a pluggable formatter, so that the pluggable formatter can serialize the object in a
selectable externalized format;
means for tracking deserialization of the object outside the pluggable formatter;
means for instantiating an uninitialized instance of an object type, so that the
object information can populate the object type; and
means for performing fixups on the objects due to forward references.
50. (Cancelled).

IX. Evidence Appendix (37 C.F.R. §41.37(c)(1)(ix))

None.

X. Related Proceedings Appendix (37 C.F.R. §41.37(c)(1)(x))

None.